Very short answer questions. "True" and "False" are considered short answers.

(1) Benchmark programs remain valid indefinitely.

(1) Is it possible to have a WAW hazard in a 5-stage MIPS pipeline?

(2) What is coherence?

(2) What is consistency?

(1) The MIPS architecture does not use condition codes.

(1) Give an example of an instruction set that is virtualizable.

(1) The design of an instruction set has no impact on how pipelinable it is.

(1) Using a different mapping scheme will reduce which type of cache miss?

(1) What concept is at the heart of RISC processing?

(1) What is the biggest problem facing MIMD processors?

(1) Which type of operation is necessary in order to support synchronization?

(1) Which hazard can be eliminated by "throwing more money at the problem"?

(1) Which type of cache miss can be reduced by using longer lines?

(1) Which type of cache miss can be reduced by using shorter lines?

(1) You can predict the cache performance of Program A by analyzing Program B.

(2) Give two techniques which will help reduce the Hit Time.

(2) Give two techniques will help reduce the Miss Penalty.

(2) Give two techniques will help reduce the Miss Rate.

(2) What is the equation to calculate the average memory access time?

(2) There are two ways to define performance - what are they?

(2) Predicting a direction is only half of the problem. What else is necessary in order to overcome control flow hazards?

(4) Cache misses can be classified into four categories. What are they?

(4) According to the Flynn taxonomy, what are the four types of parallel machines?

(3) What are the three types of hazards?

(3) Give 3 different static branch prediction techniques, in order of effectiveness.

(3) Give 3 different dynamic branch prediction techniques, in order of effectiveness.

(4) What are the four benchmark types we discussed in class?

(9) What do the following acronyms stand for?

COMA	GPGPU	EPC
SMT	UMA	VLIW
SIMD	DSM	VLSI

(3) An important program spends 50% of its time doing Floating Point operations, and 30% of its time doing integer arithmetic. By redesigning the hardware you can either make the Floating Point unit take half as long, or the integer unit 95% faster (take 5% as long). Which should you do, and why?

(2) What is the difference between an interrupt and an exception (in MIPS)?

(2) What are the two main ways to insert a stall/bubble into the pipeline?

(2) What is a "dirty bit"?

(2) Clock rates have grown by a factor of 1000 while power consumed has only grown by a factor of 30. How was this accomplished?

(2) What is the main difference between a commodity cluster and a custom cluster?

(2) Why is it difficult to come up with good benchmarks for parallel processors?

(2) What is the goal of the memory heirarchy? What two principles make it work?

(2) What is a victim cache, and how does it work?

(2) What is Strong Scaling?

(2) What is a "balanced" pipeline?

(4) The standard MIPS has a 5-stage pipeline, and uses a branch delay slot. If the machine is redesigned to be a 9-stage pipeline, with the following stages:

F D RR E1 E2 M1 M2 M3 WB (where RR stands for Register Read)

How many branch delay slots will this new design require, assuming the branch condition is calculated during E1? During E2?

(2) If we are using microprogramming in a machine with a 4-bit opcode, which has 9 control signals and 7 different states, how big is the microprogram ROM (in bits)?

(2) Assuming an 8-bit address and a 96-byte 3-way SA cache with LS=2, show how an address is partitioned/interpreted by the cache.

(2) Given 1 Megabyte of physical memory, a 22 bit Virtual address, and a page size of 8K bytes, write down the number of entries in the Page Table, and the width of each entry.

(2) What is a TLB, and what does it store?

(2) Give one other name for a non-blocking cache, and briefly explain how it works.

(2) What is Register Renaming, and why is it important?

(6) Describe the difference between shared memory and message passing machines. Include the impact on design, cost, and programming model.

(6) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why.

(4) In class, we talked about the cycle by cycle steps that occur on different interrupts. For example, here is what happens if there is an illegal operand interrupt generated by instruction i+1:

	1	2	3	4	5	6	7	8	9	
i	IF	ID	EX	MEM	WB					
i+1		IF	ID	EX	MEM	WB	<- Interrupt detected			
i+2			IF	ID	EX	MEM	WB	<- Instruction Squashed		
i+3				IF	ID	EX	MEM	WB	<- Trap Handler fetched	
i+4					IF	ID	EX	MEM	WB	

Fill out the following table if instruction i+1 experiences a fault in the Ex stage (Overflow, for example):

	1	2	3	4	5	6	7	8	9	10
i	IF	ID	EX	MEM	WB					
i+1		IF	ID	EX	MEM	WB				
i+2			IF	ID	EX	MEM	WB			
i+3				IF	ID	EX	MEM	WB		
i+4					IF	ID	EX	MEM	WB	
i+5						IF	ID	EX	MEM	WB

What happens in this case?

	1	2	3	4	5	6	7	8	9	10
i	IF	ID	EX	MEM	WB	<- Execution stage has overlflow				
i+1		IF	ID	EX	MEM	WB	<- Inst Read causes Page Fault			
i+2			IF	ID	EX	MEM	WB			
i+3				IF	ID	EX	MEM	WB		
i+4					IF	ID	EX	MEM	WB	
i+5						IF	ID	EX	MEM	WB

In this question, we are going to wire up a 16-bit version of the machine used in the midterm.

The machine has 3 different instruction formats: R, I, and J.

R-type:

Opcode	rs	rt	rd	funct
15-12	11-9	8-6	5-3	2-0

I-type:

Opcode	rs	rt	Immediate
15-12	11-9	8-6	5-0

J-type:

Opcode	Offset
15-12	11-0

The machine is byte-addressable. Immediates are sign-extended, and jumps are handled as in the MIPS processor (they are combined with the PC).

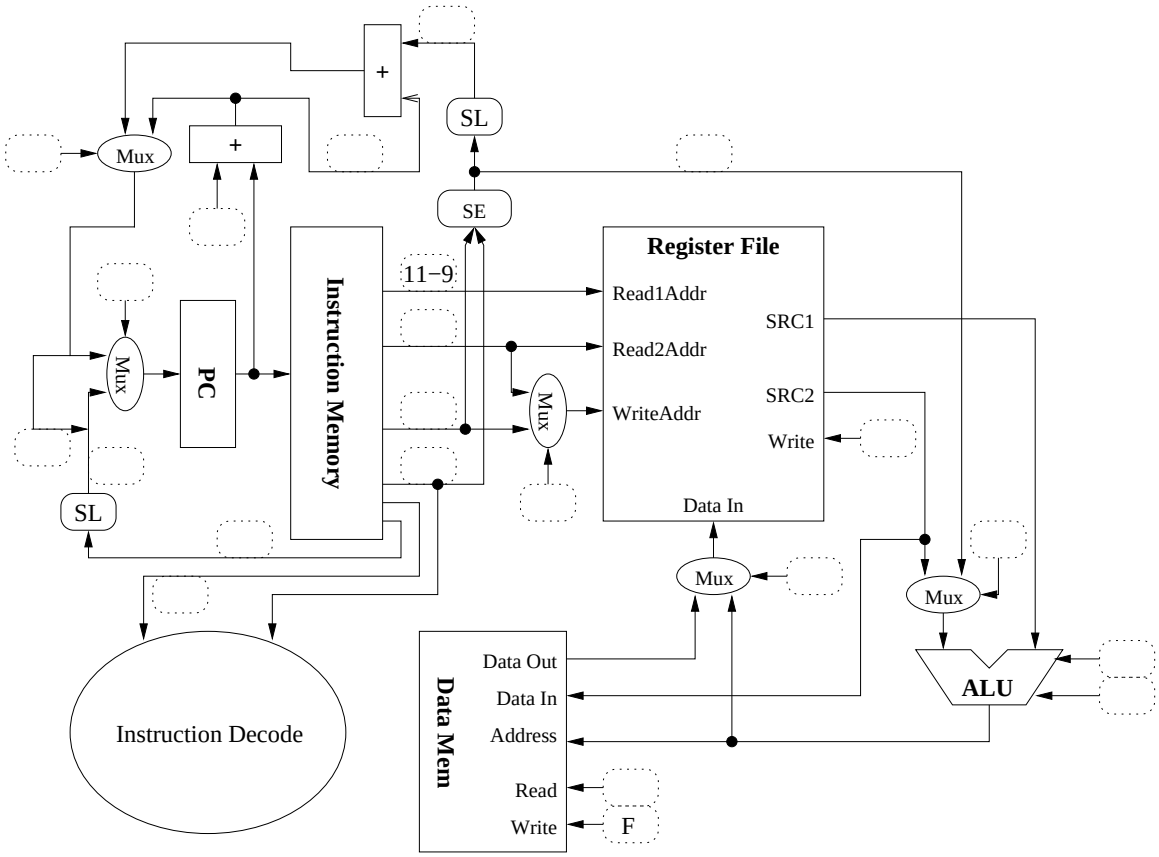
Here are some of the instructions that have been defined:

Name	**Opcode(Funct)**	**Name**	**Opcode(Funct)**	**Name**	**Opcode(Funct)**
lw	0010(xxx)	sw	0011(xxx)	NOP	0000(000)
beqz	0100(xxx)	j	0101(xxx)	NOT	1000(100)
ADD	1000(001)	ADD Imm	1001(xxx)	SUB	1000(101)
AND	1000(010)	AND Imm	1010(xxx)	XOR	1000(110)
OR	1000(011)	OR Imm	1011(xxx)	XOR Imm	1100(xxx)

Here are the 10 control signals.

A) ALU0	B) ALU1	C) ALUSrc	D) Br	E) MemRead
F) MemWrite	G) PCin	H) RegWrite	I) WrAddr	J) WrSrc

(20) In the diagram below, make sure all the dashed boxes are filled. Two are already done for you. Use the letters from the previous page instead of the actual names (F instead of MemWrite, for example).



(2) Now, write down the exact boolean equation for the MemRead signal.

(5) We need a JAL instruction - let's say it uses register 7. Sketch the changes necessary on the above diagram, and/or describe in words what would need to change in this machine in order to make this work. (Don't forget about instruction set impacts ...)

(9) In the following forwarding unit diagram, add the lines necessary to make it work.